(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 564 827 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **93103640.4**

(51) Int. Cl.5: **G06K 9/72**

(22) Date of filing: **08.03.93**

(30) Priority: **09.04.92 US 865550**

(43) Date of publication of application:
13.10.93 Bulletin 93/41

(84) Designated Contracting States:
DE FR GB

(71) Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504(US)

(72) Inventor: **Beigi, Homayoon Sadr Mohammad**
98 Front Street
Meonola, N.Y. 11501(US)

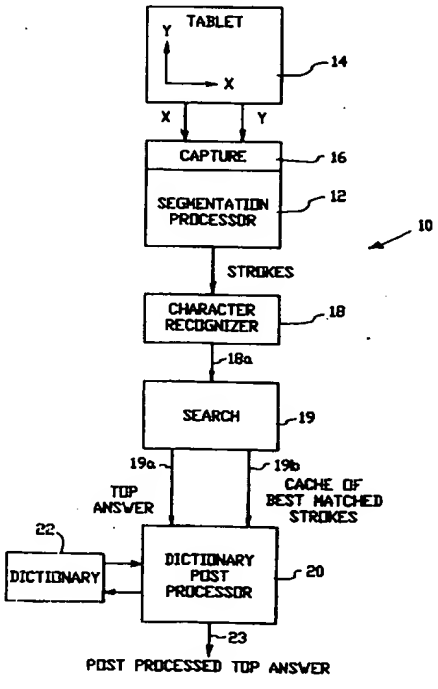
Inventor: **Fujisaki, Tetsunosuke**
4 Wayne Valley Road
Armonk, N.Y. 10504(US)
Inventor: **Modlin, William David**
255 N.E. 20th Street
Boca Raton, Fla. 33431(US)
Inventor: **Wenstrup, Kenneth Steven**
21391 Town Lakes Drive Apt. 1-19
Boca Raton, Fla. 33486(US)

(74) Representative: **Mönig, Anton, Dipl.-Ing.**
IBM Deutschland Informationssysteme GmbH,
Patentwesen und Urheberrecht
D-70548 Stuttgart (DE)

(54) A post-processing error correction scheme using a dictionary for on-line handwriting recognition.

(57) A dictionary based post-processing technique for an on-line handwriting recognition system is described. An input word has all punctuation removed, and the word is checked against a word processing dictionary. If any word matches against the dictionary, it is verified as a valid word. If it does not verify, a stroke match function and a spell-aid dictionary are used to construct a list of possible words. In some cases, the list is appended with possible words based on changing the first character of the originally recognized word. A character-match score, a substitution score and a word length are assigned to the items on the list. A word hypothesis is constructed from the list with each such word being assigned a score. The word with the best score is chosen as the output word for the processor.

FIG. 1



EP 0 564 827 A2

Field of the Invention

The invention is in the field of handwriting recognition, and specifically is directed to post-processing error correction. In particular, the error correction is accomplished using a dictionary.

BACKGROUND OF THE INVENTION

Because of similar shapes, letters such as "v" and "u"; "k" and "h"; "1", "l", and "I"; and so on, any on-line recognition of handwriting letters cannot avoid producing errors. According to the present invention, these errors and errors caused by other sources are corrected utilizing a dictionary-driven error correction post-processing technique for handwriting recognition.

Various techniques have been utilized in character recognition systems and the like which include dictionaries, but none have been found utilizing the techniques found in this invention.

U.S. Patent 4,653,107 to Shojima et al discloses a system in which coordinates of a "handwritten" pattern drawn on a tablet are sequentially sampled by a pattern "recognition" unit to prepare pattern coordinate data. Based on an area encircled by segments created by the sampled pattern coordinate data of one stroke and a line connecting a start point and an end point of the one-stroke coordinate data, the sampled pattern coordinate data of the one stroke is converted to a straight line and/or curved line segments. The converted segments are quantized and normalized. The segments of the normalized input pattern are rearranged so that the input pattern is drawn in a predetermined sequence. Differences between direction angles for the rearranged segments are calculated. Those differences are compared with differences of the direction angles of the "dictionary" patterns read from a memory to calculate a difference therebetween. The matching of the input pattern and the "dictionary" pattern is determined in accordance with the difference. If the matching fails, the first or last inputted segment of the input pattern is deleted or the sampled pattern coordinate data of the next stroke is added, to continue the "recognition" process.

U.S. Patent 5,034,991 to Hagimae et al discloses a character "recognition" method and system in which a character indicated in a printed, stamped, carved or other form is two-dimensionally imaged and stored as image data and the stored image data is subjected to an image processing to "recognize" the character. The "recognition" of the character is preformed in such a manner that each time the comparison of plural kinds of feature vectors extracted from the character to be "recognized" and a "dictionary" vector of each candidate character in a group of candidate characters preliminarily prepared is made for one of the plural kinds of feature vectors, a candidate character having its "dictionary" vector away from the extracted feature vector by a distance not smaller than a predetermined value is excluded from the candidate character group. The "dictionary" vector for each candidate character is defined as an average vector for a variety of fonts: A difference between the "dictionary" vector and the feature vector extracted from the character to be "recognized" is estimated by virtue of a deviation vector for the variety of fonts to produce an estimated value. The exclusion from the candidate character group is judged on the basis of the estimated values each of which is cumulatively produced each time the estimation for the difference is made.

U. S. Patent 5,020,117 to Ooi et al discloses a system in which "recognition" character candidates and their similarities for each character obtained by a character "recognition" section from an input character string are stored in a first "recognition" result memory, and "recognition" character candidates obtained by rotating the corresponding characters through 180 degrees and their similarities are stored in a second "recognition" result memory. Address pointers for accessing the first and second "recognition" result memories are stored in an address pointer memory. The first "recognition" result memory is accessed in accordance with the address pointers read out from the address pointer memory in an ascending order, and the second "recognition" result memory is accessed in accordance with the address pointers read out from the address pointer memory in a descending order. Coincidences between "recognition" candidates read out from the first and second "recognition" result memories and character strings of "dictionary" words read out from a "dictionary" memory are computed by a coincidence computing section. A "recognition" result of the input character string is obtained based on the coincidence.

U.S. Patent 5,010,579 to Yoshida et al discloses a hand-written, on-line character "recognition" apparatus, and the method employed by it, in which the structure of a "dictionary" for "recognition" is formed as a sub-routine type, whereby the "dictionary" can be made small in size and a time necessary for "recognition" can be reduced.

In commonly assigned U.S. Patent 5,029,223, July 2, 1991, Fujisaki discloses a method and apparatus for identifying a valid symbol or a string of valid symbols from a sequence of handwritten strokes. A method includes the steps of (a) generating in response to one or more handwritten strokes a plurality of stroke labels each having an

associated score; (b) processing the plurality of stroke labels in accordance with a beam search-like technique to identify those stroke labels indicative of a valid symbol or portion of a valid symbol; and (c) associating together identified stroke labels to determine an identity of a valid symbol or a string of valid symbols therefrom. An aspect of the invention is that each of the constraint validation filters is switchably coupled into a serial filter chain. The switches function to either couple a filter input to a stroke label or decouple the input and provide a path around the filter block. An application writer has available a plurality of constraint filters. The application writer specifies which one or ones of the constraint filters are to be applied for a specific sequence of strokes. Fujisaki is incorporated herein by reference.

As stated above, the present invention utilizes a dictionary for post-processing error correction in an on-line handwriting recognition. The just discussed patents do not teach or suggest the use of a dictionary for such a purpose.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1
is a block diagram of a dictionary based post-processor for an on-line handwriting recognition system;
FIG. 2
is a general block diagram of the dictionary post-processor of FIG. 1;
FIGS. 3A and 3B,
when taken together as shown in FIG. 3, comprise a general flow chart of the dictionary post-processor of FIG. 2;
FIGS. 4A-4F,
when taken together as shown in FIG. 4, comprises a detailed flow chart of the dictionary post-processor of FIG. 3;
FIG. 5
is a flow chart of the stroke match block 34 of FIGS. 2 and 3;
FIG. 6
is a flow chart of the spell-aid block 36 of FIGS. 2 and 3.

DISCLOSURE OF THE INVENTION

A dictionary based post-processing technique is disclosed for an on-line handwriting recognition system. An input word has all punctuation removed, and the word is checked against a word processing dictionary. If any word matches against the dictionary, it is verified as a valid word. If it does not verify, a stroke match function and a spell-aid dictionary are used to construct a list of possible words. In some cases, the list is appended with possible words based on changing the first character of the originally recognized word. A character-match score, a substitution score and a word length are assigned to the items on the list. A word hypothesis is constructed from the list with each such word being assigned a score. The word with the best score is chosen as the output word for the processor.

BEST MODE OF CARRYING OUT THE INVENTION

Referring to FIG. 1 there is shown in block diagram form a character recognition system 10 that includes a segmentation processor 12 coupled between an electronic tablet 14 and a character recognizer 18. Tablet 14 can be any of a number of suitable commercially available electronic tablets. The tablet 14 has an associated stylus or pen 15 with which, in a pen-down position, a user forms symbols, such as block printing or script alphanumeric characters, on a surface of the tablet 14. The tablet 14 has x-axis and y-axis output signals expressive of the position of the pen 15 on an x-y tablet coordinate system. A stroke capture means 16 may be a software task which intercepts the x-y outputs from the tablet to generate x-y position pair data for the segmentation processor 12. An output of the segmentation processor 12 is data expressive of connected strokes and unconnected strokes which is input to the character recognizer 18 of the invention. The character recognizer 18 operates to determine an identity of a connected group of segmented strokes and has an output 18a expressive of identified symbols such as alphanumeric characters.

In this regard it should be realized that the invention is applicable to the recognition of a number of hand-drawn symbols wherein a given symbol is composed of at least one segmented stroke. By employing the teaching of the invention the system 10 readily recognizes symbols associated with written characters of various languages and also mathematical and other types of symbols.

The output of character recognizer 18 on line 18a is provided to search block 19 which provides a top answer on line 19a and a cache of best matched strokes on line 19b. A more detailed description of blocks 14-19 can be found in Fujisaki, U.S. 5,029,233 which has been incorporated herein by reference. Dictionary post-processing is then accomplished in post-processing block 20 which compares the top answer words on line 19a with words in a dictionary 22 to produce an output word on line 23.

The top answer is a result of a search which results in the best candidate for a recognized word, and is an input word to the post-processor 20. The

cache of best matched strokes is a result of a search of the best strokes to form a word.

Refer now to FIG. 2 which is a block diagram of the dictionary post-processor 20 of FIG. 1. A punctuation filter 24 receives the top word on line 19a and the cache of different paths signal on line 19b and removes all punctuation from the word. At a verification block 26, a recognized sequence of characters is matched against the dictionary in block 28 to see if any word exists with that spelling. This match is cache insensitive. If there is a match, the word is provided to the unified cache block 30 and an output word is provided on line 31. If on the other hand there is no verification, a character match score is computed at block 32, and if the score per character is at a predetermined level, the word is output to block 30, and an output word is provided on line 31 at block 34 a stroke match is computed. Along the search process, the top match hypotheses are kept in a cache along with their scores. Basically, the objective of this module is to generate all the possible words from the strokes in this cache and to calculate the total matching score of each of these words. Then the end word hypotheses with the best matching scores are inserted into a global word hypotheses list. The output of the match 34 is then provided to a spell-aid block 36 which takes the recognition output and tries to find a word in the dictionary which resembles this sequence of characters and inserts them in the global word hypotheses list. This model's output is very dependent on the initial characters. Therefore, most of the time, the first character is retained. Therefore, if the match score of the first character is worse than other characters in the word, then it is replaced in block 38. At block 40, three types of scores are assigned to the replacement characters, and at block 42, the best hypothesis in the list is determined. At block 44, the best hypothesis is used as the final word, and punctuation is reinserted at block 46 with an output word being provided on line 48.

Refer now to FIG. 3 which is a more detailed block diagram of the dictionary post-processor 20. In FIG. 3A, at block 24, the top answer is provided on input line 19a and the cache of best matched strokes signal is provided on line 19b, and the punctuation is removed from the top answer input at block 24. At decision block 26, a determination is made whether the word is verified in the dictionary. If so, the word is provided to the unified cache block 30 and an output word is provided on line 31. If on the other hand, the word does not exist in the dictionary, proceed to character match score block 32 which is comprised of blocks 50 and 52. In block 50, scores are calculated for each character in the top answer word. At decision block 52, a determination is made whether or not the worse character score is better than a predetermined threshold. If so, proceed to block 30 to unify the case and provide an output word on line 31. If not, proceed to block 34 where a stroke match is made using the cache to find all combinations of strokes which will verify. At spell-aid block 36, a standard word processor dictionary is used to get some suggested words.

Proceed next to first character replacement block 38 in FIG. 3B which is comprised of blocks 54 and 56. In decision block 54, a determination is made whether or not the first character in the top answer has the worst character match score among all the characters in the top answer. If so, proceed to block 56 and get a hypothesis by changing the first character using statistics of first characters in a word. Proceed then to block 40 and assign a character-match score to the top answer and substitution scores and word length scores to all hypotheses. In block 42, find the hypothesis with the best of all relative scores based on the following precedent: 1. word length 2. substitution 3. relative character-match. After this determination, proceed to block 44 and unify the case with the hypothesized word which is then provided to block 46 where punctuation is reinserted and an output word is provided on line 48.

Refer now to FIG. 4. FIG. 4 which is a detailed flow chart of the operation of the post-processor block 20. The flow chart starts at block 60 of FIG. 4A, and at block 62, a sequence of characters is extracted from the top search path and are stored in "word" and "original word". At block 64, all punctuation is removed from the "word" and the punctuation is stored. At block 66, a case insensitive match of "word" is made against the dictionary database. If "word" is made up of non-alphabetical characters only, then it is verified. If it has any special characters such as punctuation marks etc, they are separated and kept in a separate portion called "punctuation". At decision blocks 68, a determination is made if the "word" is verified. If so, proceed to block 70 and if the first character of the "original word" is upper case, retain its case. Count the number of lower and upper case characters and convert all character cases to the majority case in the "original word". The "original word" is then provided as an output word on line 72.

If in decision blocks 68 the word is not verified, go to block 74 of FIG. 4B where the shape matching score for each character in the word is looked up. At decision block 76, it is determined if the sum of scores is less then a threshold times the length of the "word". If so, return to block 70 (FIG. 4A) and generate an output word on line 72. If the determination is that the sum of scores is not less than the threshold times the length of the word, proceed to block 78 where a linear transformation

of the scores is performed such that the highest score is mapped to zero and zero is mapped to the highest score. Call the new scores "character-match" scores. For those characters in the "word" which have no match scores associated with them, assign a character-match score of "-1" which is worse than all other scores. At block 80, for the word hypothesis given by stroke-match and spell-aid, if their characters have a match score associated, then transform those match-scores using the above linear transformation and assign these scores to those characters as the character-match score. If no match score is available, then use "0" as their character-match score. At block 82, get a list of suggested words from the stroke match, and proceed to block 84 and append to this list a list of words suggested by the spell-aid.

At blocks 86 of FIG. 4C, if the first character in "word" has transformed score "0" or "-1", then try changing the first character with other characters given through a study of the probability of characters occurring in the beginning of a word from a predetermined word corpus, for example, 320,000,000 containing a predetermined number of distinct words, for example, 270,000 such words. At block 88, for each word hypothesis given by stroke-match and spell-aid, insert the number of substitutions that will make the word compared to the original word. Call this "substitution score" (SS). Proceed then to decision block 90 for the determination of whether or not this is a strong dictionary. If not, proceed to block 92 of FIG. 4C where L equals the length of the "original word" and SS equals the substitution score. At decision block 94, a test is made to keep the error correction robust.

If not robust, proceed to block 96 and set SS equal to -1 and proceed to block 98 of FIG. 4D and for every word hypothesized by stroke-match and spell-aid, find the difference in their length with "word". This same path is taken if there is a strong dictionary decision at block 90. Be careful with presence of punctuations. Call this the "word-length" score. At block 100, loop through all word hypotheses and their scores. At block 102 find all hypotheses with the smallest word-length score. At block 104, find those hypotheses with the smallest substitution scores. Proceed to block 106 of FIG. 4E and among these word hypotheses find the hypotheses which tend to make the most substitutions for the position of characters which had a character match score of "-1" in "word". At block 108 among the remaining hypotheses, find that hypothesis which has the smallest sum of absolute values of the difference of the character-match scores of "word" and this hypothesis is kept. In block 110, if the remaining list of hypotheses has more than one element, then keep that hypotheses which originated from the stroke match module.

At decision block 112 of FIG. 4E, a determination is made if the first character of the "original word" is upper case. If not, go to block 114 and set the upper case equal to zero, and proceed to block 118. If so, proceed to block 116 and set the upper case equal to one. Proceed then to decision block 118 where a determination is made whether or not most of the characters in the "original word" are lower case. If so, proceed to block 120 and set the lower case equal to one and then proceed to block 124. If in decision block 118 most of the characters in the original word are not lower case proceed to block 122 and set lower case equal to zero. At decision block 124, a determination is made if the lower case equals zero. If not, proceed to block 126 and turn all characters in the hypotheses into lower case. If lower case equals to zero in decision block 124, proceed to block 134 and turn all characters in the hypothesis into upper case, and then proceed to block 132 at FIG. 4F. At block 128 of FIG. 4F, a determination is made if the upper case equals one, if not, proceed to block 132. If so, proceed to block 130 and turn the first character in the hypothesis into upper case. Proceed then to block 132 and copy the hypotheses into the original word. At block 136 punctuation is reinserted in the word and an output word is provided on line 138.

Refer now to FIG. 5 which is a detailed flow chart of the stroke match block 34 shown in FIGS. 2 and 3. The flow chart begins at 140. At block 142, the top score stroke hypothesis is taken from the stroke matcher and all combinations of strokes are found which make valid words in the dictionary. At block 144 all these scores for the strokes in each word hypotheses are added. At block 146 a list of "N" hypotheses with the best total score is made and a return list is made at block 148.

Refer now to FIG. 6 which is a detailed flow chart of the spell-aid block 36 of FIGS. 2 and 3. The flow chart is started at block 150, and at block 152, the "word" is passed to word processor spell checker to obtain the first six words which most resemble the "word" and these are returned to the list at block 154.

INDUSTRIAL APPLICABILITY

It is an object of the invention to provide an improved handwriting recognition system.

It is an object of the invention to provide an improved handwriting recognition system utilizing a dictionary based post-processing technique.

Claims

1. A method of using a dictionary for on-line hand-writing recognition, said method comprising the steps of:
providing a candidate word for recognition, where said candidate word is made up of a sequence of at least one character which is made up of a sequence of at least one stroke;
determining if the sequence of characters in said candidate word matches a word in the dictionary with the same spelling, and if so, providing the candidate word as an output word; and if not
calculating a recognition score for each character in said candidate word;
determining if the worst character score in said recognition score for each character is better than a predetermined threshold, and if so, providing said candidate word as an output word; and if not
finding all combinations of strokes that produce a recognizable character to be used in place of the character with the worst character score;
assigning scores to each of the recognizable characters;
replacing the character with the worst character score in said candidate word with the recognizable character with the highest assigned score to produce a new candidate word; and
providing said new candidate word as an output word.

2. The method of claim 1, including the step of:
removing punctuation from said candidate word.

3. The method of claim 1 or 2, including the step of:
inserting the punctuation removed from said candidate word in said provided candidate word.

4. The method of claim 1, 2 or 3, including the step of:
inserting the punctuation removed from said candidate word in the new candidate word.

FIG. 1

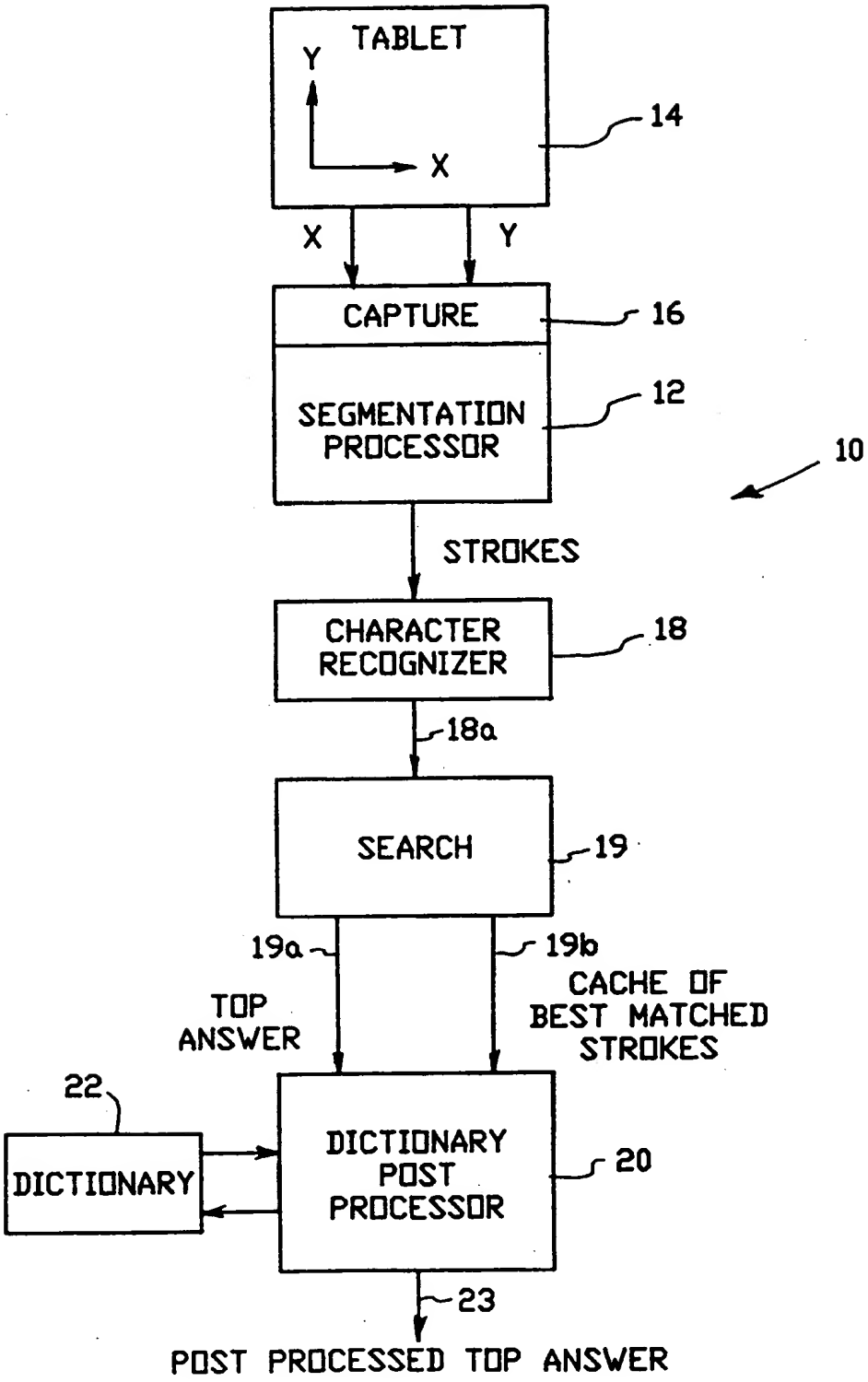
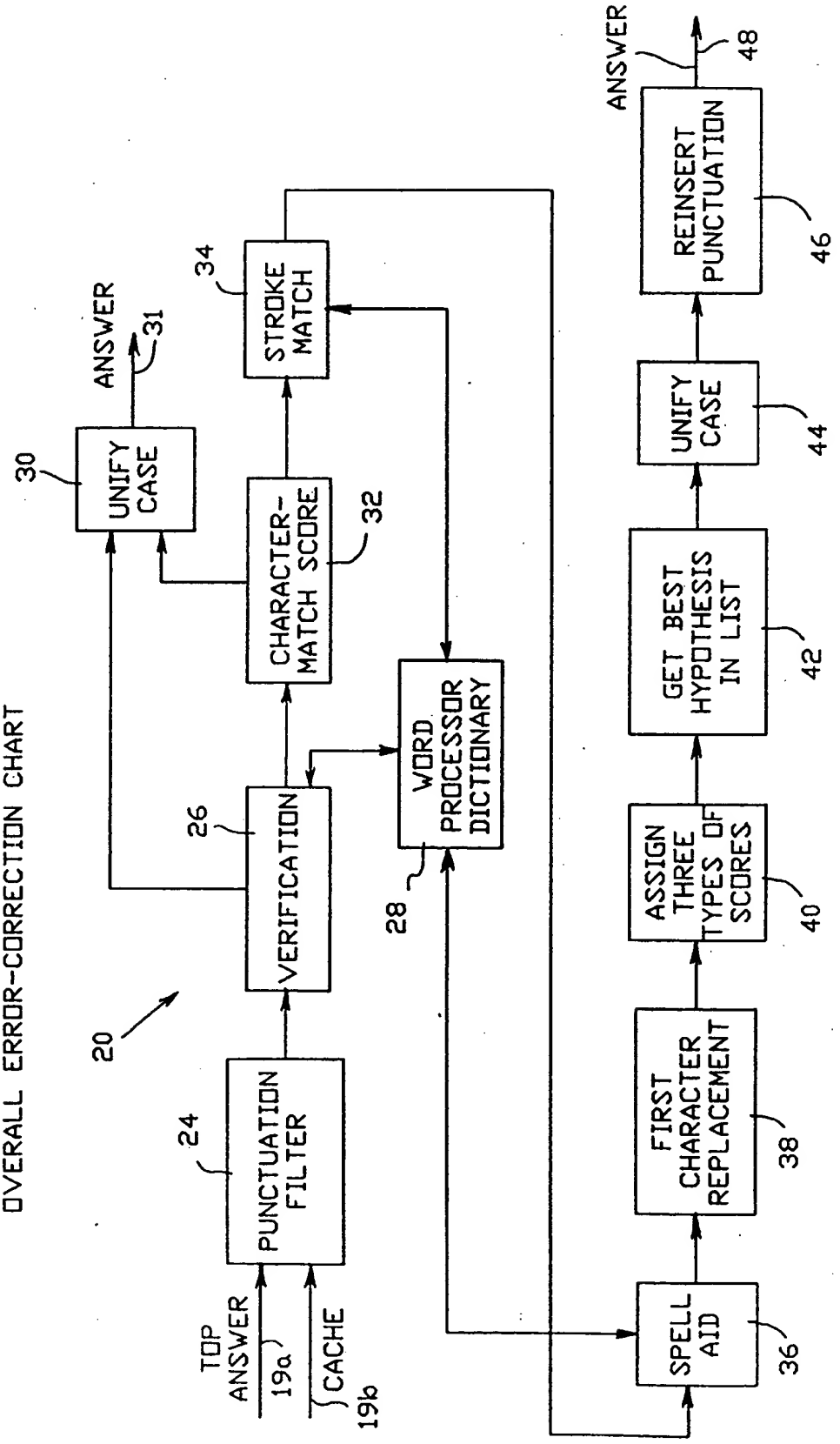

TABLET
Y
X
14
X
Y
CAPTURE
16
SEGMENTATION PROCESSOR
12
10
STROKES
CHARACTER RECOGNIZER
18
18a
SEARCH
19
19a
19b
TOP ANSWER
CACHE OF BEST MATCHED STROKES
22
DICTIONARY
DICTIONARY POST PROCESSOR
20
23
POST PROCESSED TOP ANSWER

FIG. 2

OVERALL ERROR-CORRECTION CHART



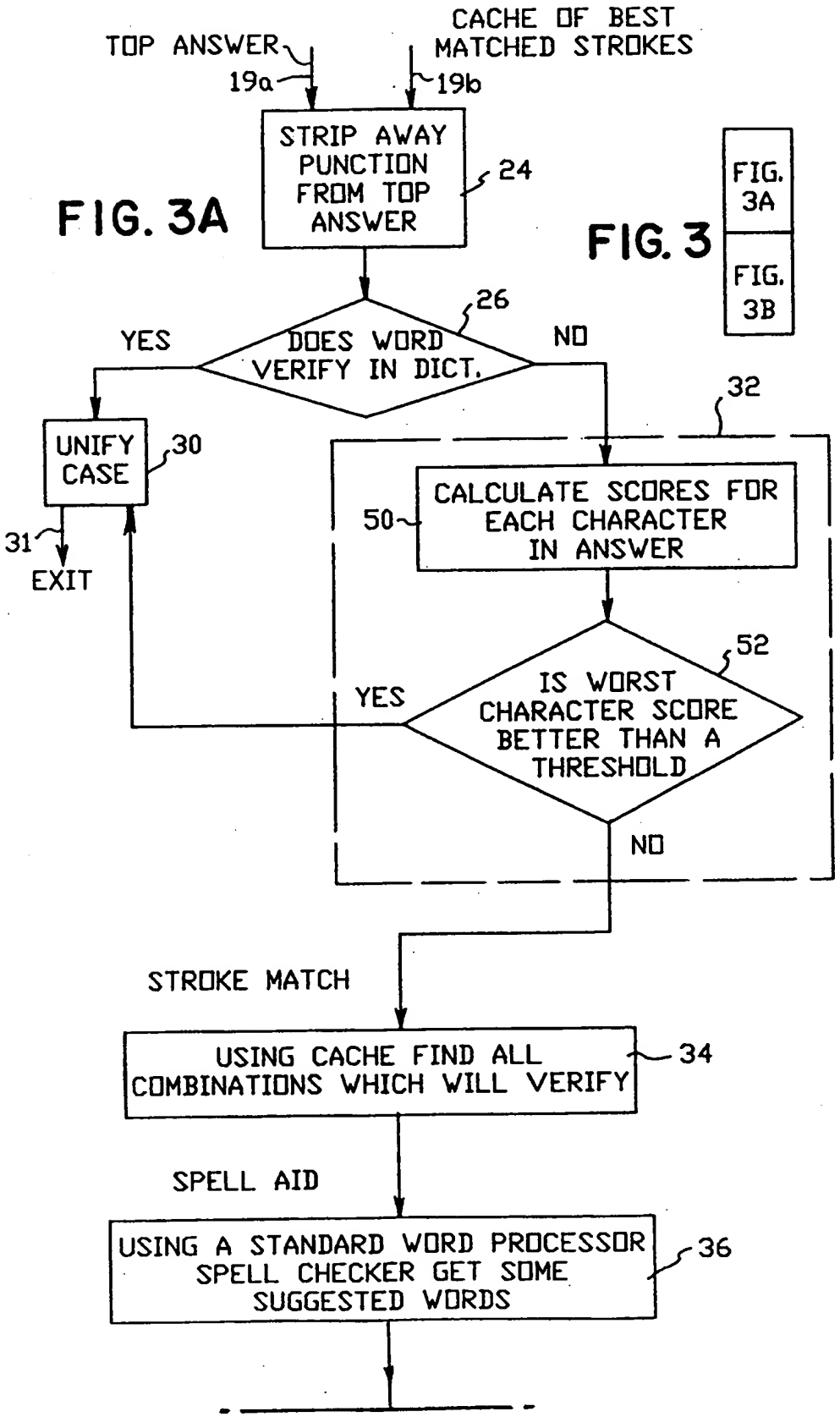


FIG. 3A

FIG. 3

FIG. 3A
FIG. 3B

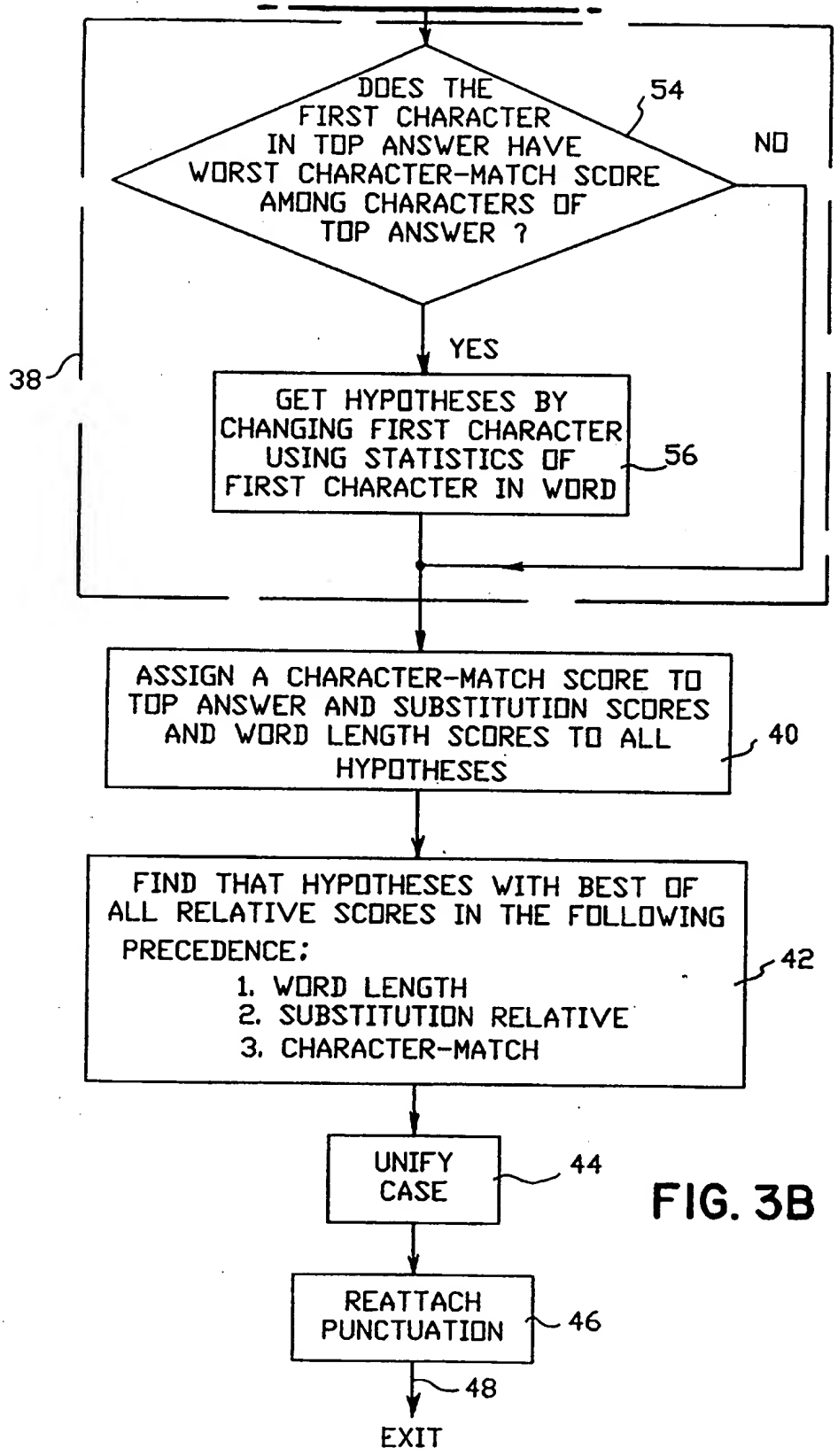

DOES THE FIRST CHARACTER IN TOP ANSWER HAVE WORST CHARACTER-MATCH SCORE AMONG CHARACTERS OF TOP ANSWER ?
54
NO
YES
38
GET HYPOTHESES BY CHANGING FIRST CHARACTER USING STATISTICS OF FIRST CHARACTER IN WORD
56
ASSIGN A CHARACTER-MATCH SCORE TO TOP ANSWER AND SUBSTITUTION SCORES AND WORD LENGTH SCORES TO ALL HYPOTHESES
40
FIND THAT HYPOTHESES WITH BEST OF ALL RELATIVE SCORES IN THE FOLLOWING PRECEDENCE:
1. WORD LENGTH
2. SUBSTITUTION RELATIVE
3. CHARACTER-MATCH
42
UNIFY CASE
44
REATTACH PUNCTUATION
46
48
EXIT

FIG. 3B

FIG. 4A

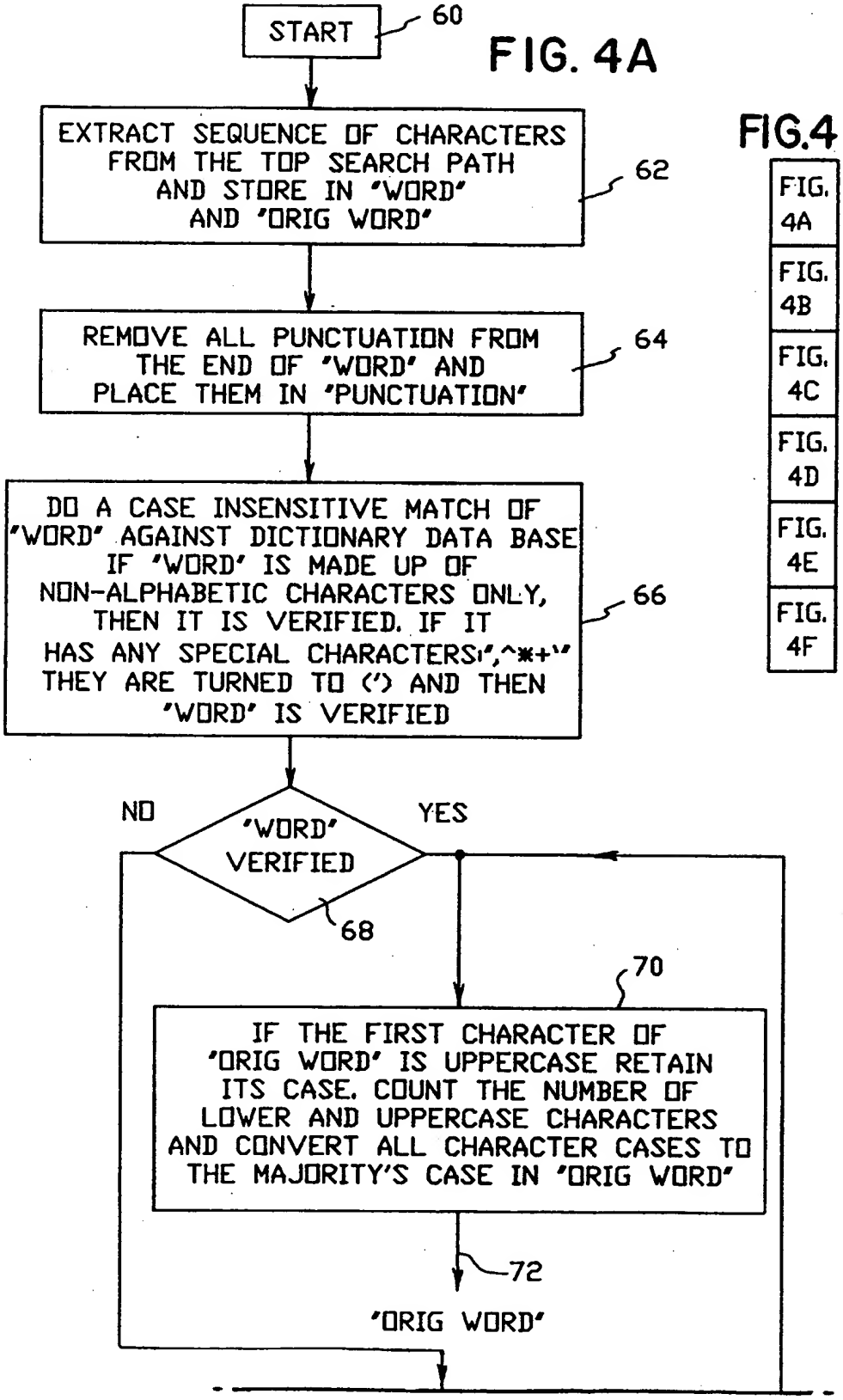
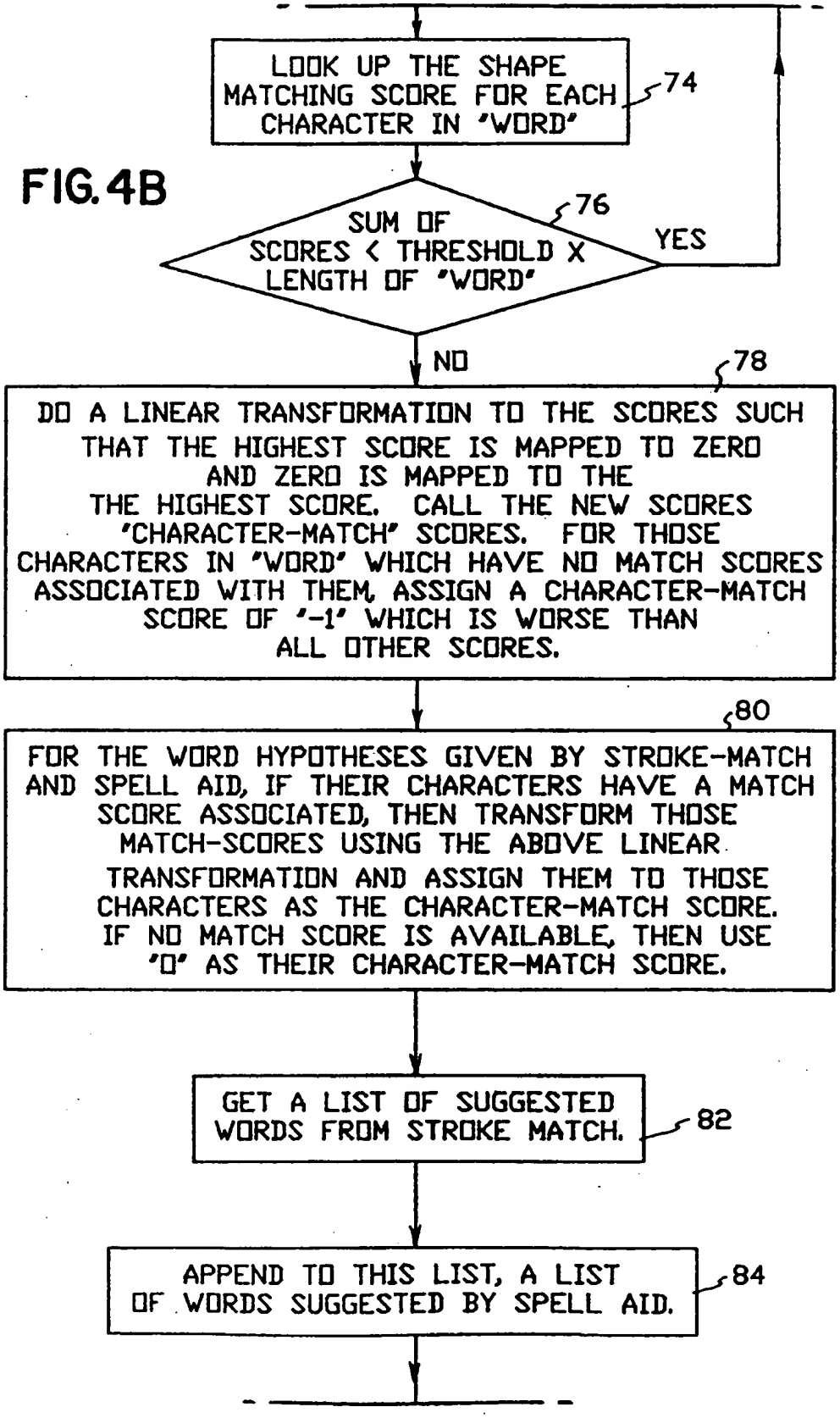


FIG.4

FIG. 4A
FIG. 4B
FIG. 4C
FIG. 4D
FIG. 4E
FIG. 4F

FIG. 4B


LOOK UP THE SHAPE MATCHING SCORE FOR EACH CHARACTER IN "WORD"
74
SUM OF SCORES < THRESHOLD X LENGTH OF "WORD"
76
YES
NO
78
DO A LINEAR TRANSFORMATION TO THE SCORES SUCH THAT THE HIGHEST SCORE IS MAPPED TO ZERO AND ZERO IS MAPPED TO THE THE HIGHEST SCORE. CALL THE NEW SCORES "CHARACTER-MATCH" SCORES. FOR THOSE CHARACTERS IN "WORD" WHICH HAVE NO MATCH SCORES ASSOCIATED WITH THEM, ASSIGN A CHARACTER-MATCH SCORE OF "-1" WHICH IS WORSE THAN ALL OTHER SCORES.
80
FOR THE WORD HYPOTHESES GIVEN BY STROKE-MATCH AND SPELL AID, IF THEIR CHARACTERS HAVE A MATCH SCORE ASSOCIATED, THEN TRANSFORM THOSE MATCH-SCORES USING THE ABOVE LINEAR TRANSFORMATION AND ASSIGN THEM TO THOSE CHARACTERS AS THE CHARACTER-MATCH SCORE. IF NO MATCH SCORE IS AVAILABLE, THEN USE "0" AS THEIR CHARACTER-MATCH SCORE.
GET A LIST OF SUGGESTED WORDS FROM STROKE MATCH.
82
APPEND TO THIS LIST, A LIST OF WORDS SUGGESTED BY SPELL AID.
84

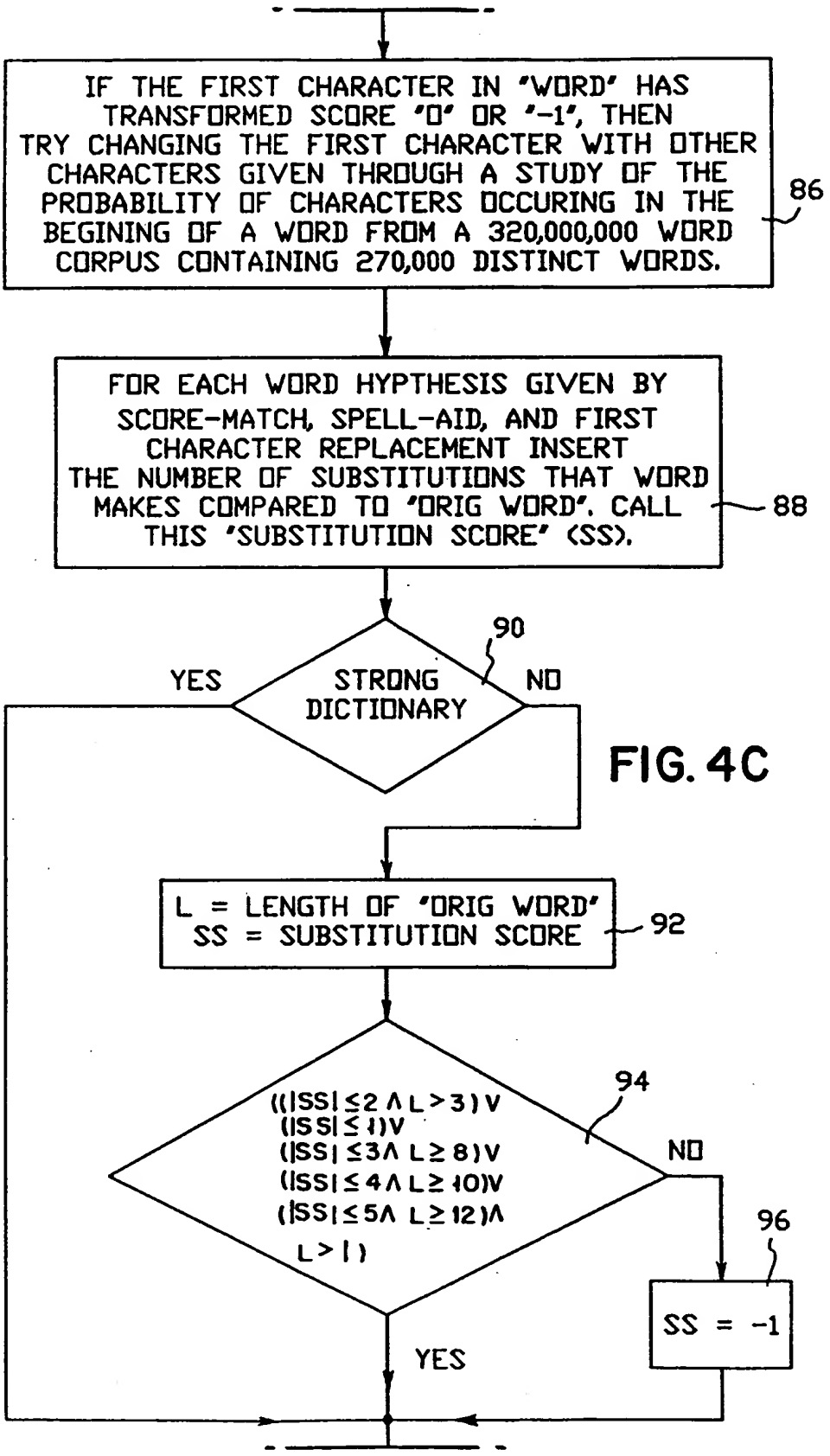

IF THE FIRST CHARACTER IN 'WORD' HAS TRANSFORMED SCORE '0' OR '-1', THEN TRY CHANGING THE FIRST CHARACTER WITH OTHER CHARACTERS GIVEN THROUGH A STUDY OF THE PROBABILITY OF CHARACTERS OCCURING IN THE BEGINING OF A WORD FROM A 320,000,000 WORD CORPUS CONTAINING 270,000 DISTINCT WORDS.
86
FOR EACH WORD HYPTHESIS GIVEN BY SCORE-MATCH, SPELL-AID, AND FIRST CHARACTER REPLACEMENT INSERT THE NUMBER OF SUBSTITUTIONS THAT WORD MAKES COMPARED TO 'ORIG WORD'. CALL THIS 'SUBSTITUTION SCORE' (SS).
88
90
YES
STRONG DICTIONARY
NO
L = LENGTH OF 'ORIG WORD'
SS = SUBSTITUTION SCORE
92
94
((|SS| ≤ 2 ∧ L > 3) ∨
(|SS| ≤ 1) ∨
(|SS| ≤ 3 ∧ L ≥ 8) ∨
(|SS| ≤ 4 ∧ L ≥ 10) ∨
(|SS| ≤ 5 ∧ L ≥ 12) ∧
L > 1)
NO
96
SS = -1
YES

FIG. 4C

FIG. 4D

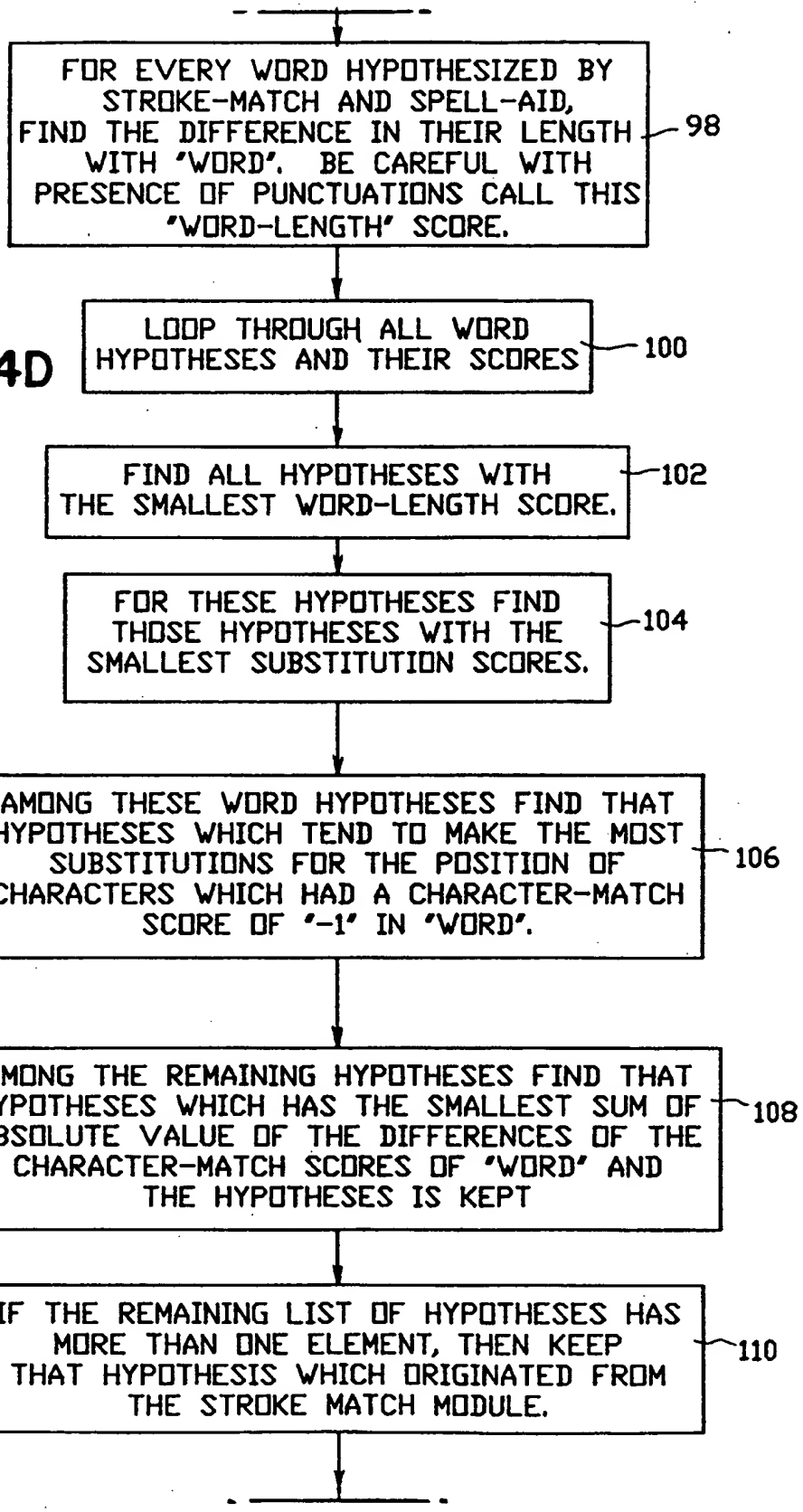

FOR EVERY WORD HYPOTHESIZED BY STROKE-MATCH AND SPELL-AID, FIND THE DIFFERENCE IN THEIR LENGTH WITH 'WORD'. BE CAREFUL WITH PRESENCE OF PUNCTUATIONS CALL THIS 'WORD-LENGTH' SCORE.
98
LOOP THROUGH ALL WORD HYPOTHESES AND THEIR SCORES
100
FIND ALL HYPOTHESES WITH THE SMALLEST WORD-LENGTH SCORE.
102
FOR THESE HYPOTHESES FIND THOSE HYPOTHESES WITH THE SMALLEST SUBSTITUTION SCORES.
104
AMONG THESE WORD HYPOTHESES FIND THAT HYPOTHESES WHICH TEND TO MAKE THE MOST SUBSTITUTIONS FOR THE POSITION OF CHARACTERS WHICH HAD A CHARACTER-MATCH SCORE OF '-1' IN 'WORD'.
106
AMONG THE REMAINING HYPOTHESES FIND THAT HYPOTHESES WHICH HAS THE SMALLEST SUM OF ABSOLUTE VALUE OF THE DIFFERENCES OF THE CHARACTER-MATCH SCORES OF 'WORD' AND THE HYPOTHESES IS KEPT
108
IF THE REMAINING LIST OF HYPOTHESES HAS MORE THAN ONE ELEMENT, THEN KEEP THAT HYPOTHESIS WHICH ORIGINATED FROM THE STROKE MATCH MODULE.
110

FIG. 4E

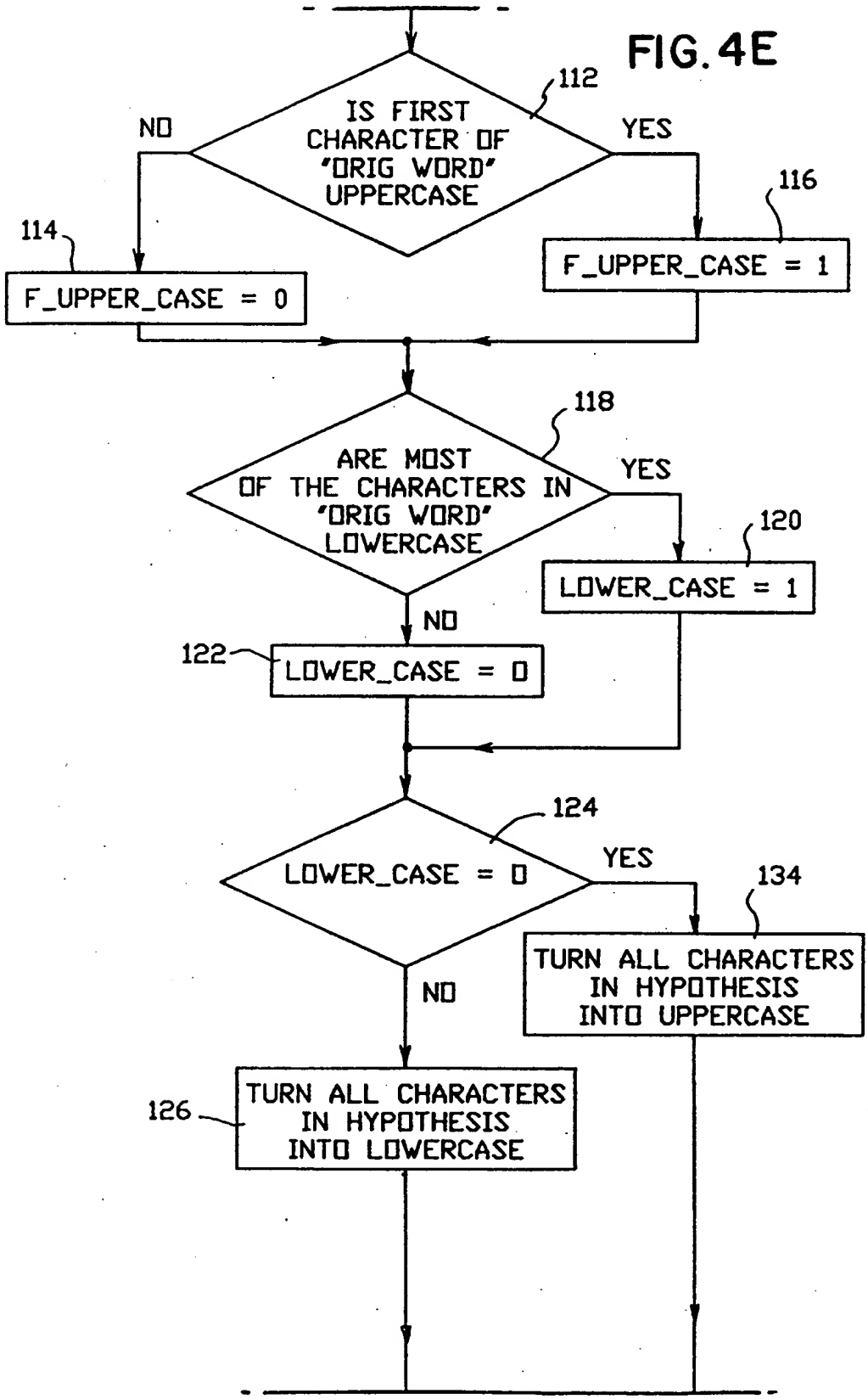

IS FIRST CHARACTER OF "ORIG WORD" UPPERCASE
112
NO
YES
114
F_UPPER_CASE = 0
116
F_UPPER_CASE = 1
ARE MOST OF THE CHARACTERS IN "ORIG WORD" LOWERCASE
118
YES
120
LOWER_CASE = 1
NO
122
LOWER_CASE = 0
LOWER_CASE = 0
124
YES
134
TURN ALL CHARACTERS IN HYPOTHESIS INTO UPPERCASE
NO
126
TURN ALL CHARACTERS IN HYPOTHESIS INTO LOWERCASE

FIG. 4F

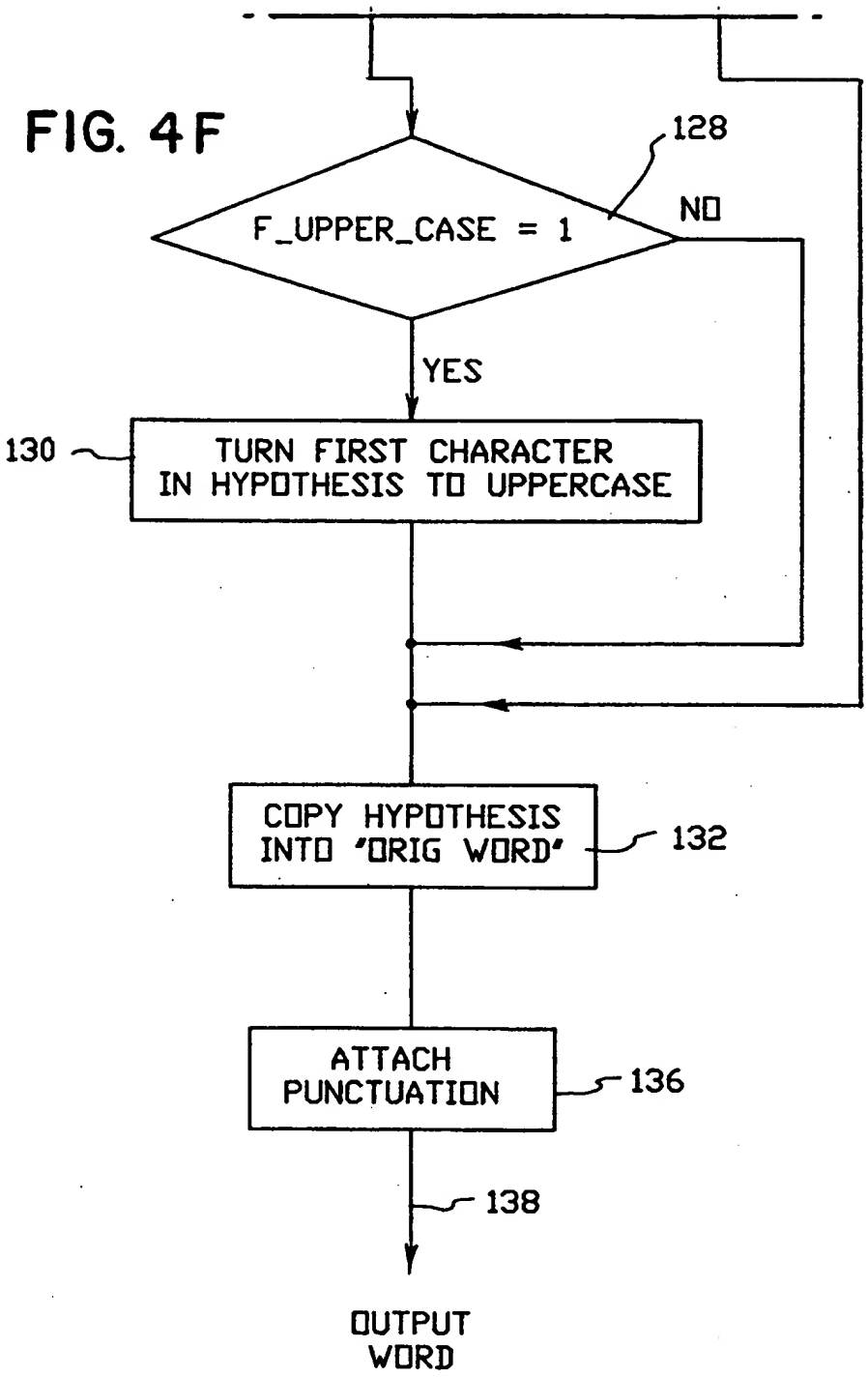

128
F_UPPER_CASE = 1
NO
YES
130
TURN FIRST CHARACTER IN HYPOTHESIS TO UPPERCASE
COPY HYPOTHESIS INTO 'ORIG WORD'
132
ATTACH PUNCTUATION
136
138
OUTPUT WORD

FIG. 5

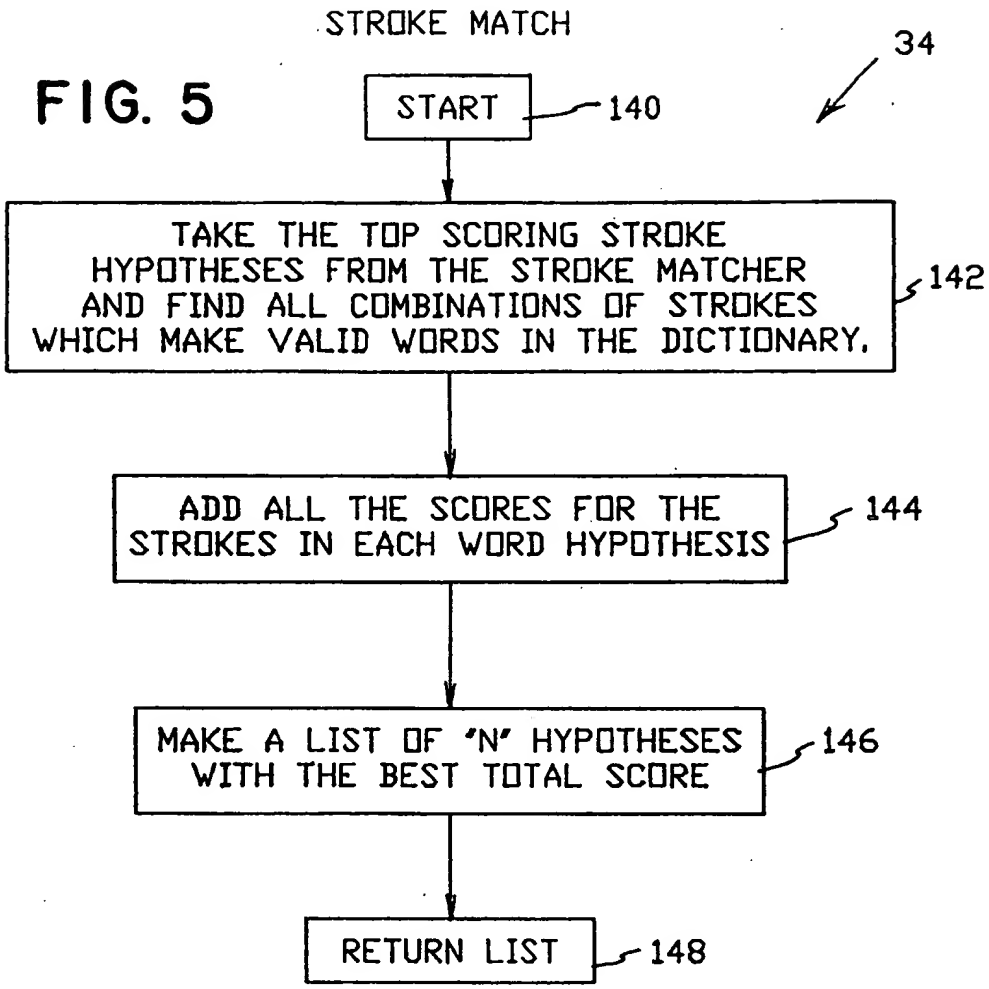
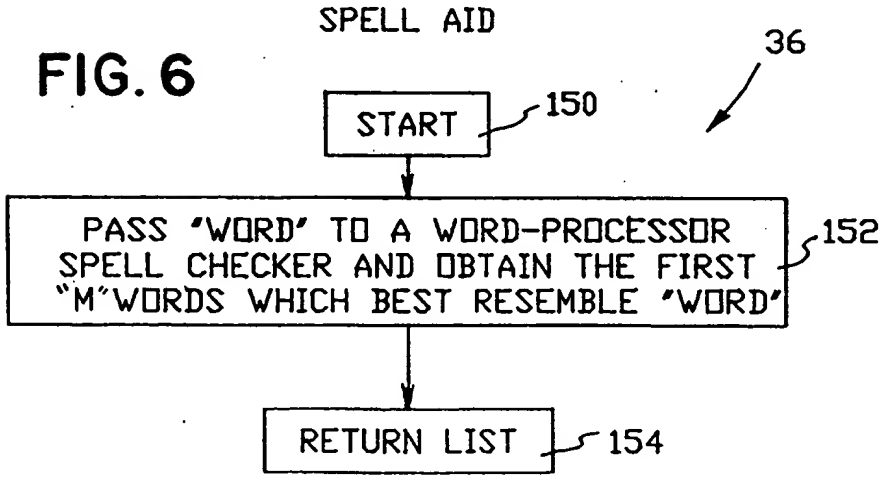

STROKE MATCH
34
START
140
TAKE THE TOP SCORING STROKE HYPOTHESES FROM THE STROKE MATCHER AND FIND ALL COMBINATIONS OF STROKES WHICH MAKE VALID WORDS IN THE DICTIONARY.
142
ADD ALL THE SCORES FOR THE STROKES IN EACH WORD HYPOTHESIS
144
MAKE A LIST OF "N" HYPOTHESES WITH THE BEST TOTAL SCORE
146
RETURN LIST
148

FIG. 6


SPELL AID
36
START
150
PASS "WORD" TO A WORD-PROCESSOR SPELL CHECKER AND OBTAIN THE FIRST "M" WORDS WHICH BEST RESEMBLE "WORD"
152
RETURN LIST
154